

Embodiments of the disclosed methods and systems, such as Resource Model Equations (4), (7) and (8A), may be implemented with a variety of information management system I/O configurations. For example, the disclosed resource models described above may be implemented as I/O admission control policies that act as a "soft control" to I/O scheduling. In addition, it is also possible that various I/O scheduling algorithms may be applied in conjunction with the disclosed I/O admission control policies. For example, the disclosed methods and systems may be implemented with earliest-deadline-first scheduling schemes as described in A. Reddy and J. Wylie, "I/O Issues in a Multimedia System", IEEE Computer, 27(3), pp. 69-74, 1994; and R. Haskin, "Tiger Shark --- A Scalable File System for Multimedia", IBM Journal of Research and Development, Vol. 42, No. 2, pp. 185-198, March 1998, each of which is incorporated herein by reference. In such a case, the deadline of an I/O request may be calculated, for example, based on cycle time T and consumption rate  $P_i$ .

The disclosed methods and systems may also be implemented in systems employing both lower level admission control policies and round-based disk arm scheduling techniques. Examples of conventional storage system configurations employing round-based scheduling techniques and admission control policies (e.g., SCAN), are described in T. Teorey and T. Pinkerton, "A comparative analysis of disk scheduling policies", Communications of the ACM, 15(3), pp. 177-184, 1972, which is incorporated herein by reference. In such conventional implementations, it is typically desirable that lower level admission control policies be closely coupled with the disk arm scheduling algorithm. Although it is possible to modify the disclosed resource model embodiments (e.g., Resource Model Equations (4), (7) and (8A)) to reflect disk arm scheduling factors, this is not necessary in the logical volume management level. Instead, average access AA and average transfer rate TR may be relied on to factor in the impacts of disk arm level details. Advantageously, this allows lower level implementation details to be transparent.

Embodiments of the disclosed methods and systems may also be implemented with a variety of information management system operations. For example, Resource Model Equations (4), (7) and (8A) may be implemented in information management system operations including,

but not limited to, read-only activities for video streams. Such activities typically comprise the major portion of I/O workloads for information management systems, such as content delivery systems, content router systems, *etc.* However, the disclosed methods and systems may also be implemented in conjunction with other types or classes of I/O activities, including background system I/O activities such as the writing of large video files to system storage devices (*e.g.*, when updating content on one or more storage devices), and/or for the accessing of relatively small files (*e.g.*, metadata files, index files, *etc.*). When implemented in conjunction with such activities, the disclosed methods and systems may be modified so as to consider workload demands particular to these types of I/O activities, for example in a manner as follows.

In one embodiment, the disclosed admission control policies may be implemented in a manner that addresses writing operations for large continuous file (*e.g.*, writing of relatively large video files as part of controlled or scheduled content provisioning activity), and/or access operations for relatively small files. Writing operations for relatively large files may occur at any time, however, it is common to attempt to schedule them during maintenance hours or other times when client demand is light. However, it is also possible that such writing operations may occur during primary system run-time when client demand is relatively heavy, *e.g.*, when a remote copy is downloaded to a local server. Furthermore, even in the case of file writing operations scheduled during maintenance time windows, the workload for writing relatively large files may consume a significant portion of buffer space and I/O capacity for a given period of time, and especially in the event that client demand surges unexpectedly when the system is updating its contents. Small file access may not necessarily consume a lot of I/O resources, but may have higher timing requirements. Such small files may contain critical information (*e.g.*, metadata, index file data, I-node/D-node data, overhead data for stream encoding/decoding, *etc.*), so that I/O demand for these tasks should be served as soon as possible.

Under the above-described conditions of heavy resource demand and/or critical timing for access, embodiments of the disclosed methods and systems may be implemented to provide a resource manager that at least partially allocates information management system I/O resources among competing demands, for example, by using the management layer to define a certain portion or percentage of I/O capacity, and/or certain portion or percentage of buffer space,

allowed to be used for writing operations such as content updating. In such embodiments, the allocated portion of either I/O capacity or buffer space may be fixed, may be implemented to vary with time (*e.g.*, in a predetermined manner, based on monitored information management system I/O resources/characteristics, *etc.*), may be implemented to vary with operator input via a storage management system, *etc.*

In one embodiment, resource utilization balance may be maintained by reserving a fixed or defined portion of cycle time T to be utilized for content-updating/content provisioning workloads. In this embodiment, the reserved portion may be configurable on a real-time basis during runtime, and/or be made to vary with time, for example, so that the portion of T reserved for content-updating may be higher when an information management system is in maintenance mode, and lower when the system is in normal operational mode. In one exemplary embodiment, a configurable resource parameter (*e.g.*, "Reserved\_Factor") having a value of from about 0 to about 1 may be employed to reflect portion or percentage of I/O resources allocated for internal system background processing activity (*e.g.*, large file updates, small file accesses, *etc.*). In this embodiment, the balance of I/O resources (*e.g.*, "1 - Reserved\_Factor") may be used for processing the admission of new viewers.

A configurable resource parameter such as Reserved\_Factor may be fixed in value (*e.g.*, a predetermined value based on estimated processing background needs), or may be implemented to vary with time. A variable resource parameter may be implemented using at least two parameter values (*e.g.*, at least two constant values input by operator and/or system manager) that vary according to a predetermined schedule, or may be a dynamically changing value based at least in part on monitored information management system resources/characteristics, such as monitored background system processing activity. For example, a first value of Reserved\_Factor may be set to be a predetermined constant (*e.g.*, about 0.1) suitable for handling normal processing background activities at times of the day or week during which an information management system workload is anticipated to include primarily or substantially all read-only type activities for video streams. A second value of Reserved\_Factor may be set to be a predetermined constant (*e.g.*, about 0.3) suitable for handling content provisioning processing